

UNIVERSITA' DI GENOVA

Istituto di Meccanica Applicata alle Macchine

Rapporto IIR.70.AL.145

CALCOLATORI IBRIDI E LINGUAGGI DI SIMULAZIONE NELLO STUDIO DELLA DINAMICA

DEI SISTEMI CONTINUI

To be published by "Automazione e Automatismi"

(NASA-CR-127252) HYBRID COMPUTERS AND SIMULATION LANGUAGES IN THE STUDY OF DYNAMICS OF CONTINUOUS SYSTEMS G.M. Acaccia, et al (Genoa Univ.) Oct. 1970 12 p	N72-27220  Unclas G3/08 90002
--	--



FRESK

Dr. G.M.Acaccia

Dr.Ing. A.L.Lucifredi

Genova, Ottobre 1970

ESRO, follow

12p

# CALCOLATORI IBRIDI E LINGUAGGI DI SIMULAZIONE NELLO STUDIO DELLA DINAMICA DEI SISTEMI CONTINUI

## Sommario

La nota prende in esame i due approcci che consentono lo studio diretto, per simulazione, del comportamento dei sistemi dinamici; e cioè:

l'impiego dei calcolatori ibridi

l'impiego dei linguaggi di simulazione

Definite entrambe le procedure e richiamati per ciascuna pregi e difetti allo stato attuale delle conoscenze e delle apparecchiature, vengono formulati criteri di confronto e di valutazione reciproca.

## Introduzione

In questi ultimi tempi si è venuta delineando l'esigenza di una sempre più completa conoscenza del comportamento dinamico di sistemi (o). Attualmente lo studio può essere fatto con diverse procedure aventi un carattere fisico-sperimentale oppure pervenendo ad una identificazione di tale sistema con un modello matematico mediante una serie di variabili indipendenti e dipendenti che caratterizzino il processo in esame, legate tra loro da relazioni di tipo matematico o logico ed, una volta costruito tale modello, simulando il sistema in esame su elaboratori digitali, calcolatori analogici, sistemi ibridi analogico-digitale, per giungere infine ad una previsione del loro comportamento.

Quest'ultimo procedimento è particolarmente utile, perchè consente lo studio di sistemi anche molto complessi, la cui costruzione sotto forma di modello fisico si presenta molto difficile o quasi impossibile e comunque molto costosa e la caratterizzazione puramente analitica del modello matematico estremamente complessa.

Va osservato inoltre che la simulazione per il tramite di un modello matematico, benchè come procedura non lo ottimizzi, ne consente tuttavia l'affinamento, essendo un processo di tipo ripetitivo; questo obiettivo è naturalmente assai più difficile da raggiungere su un modello fisico.

Il sistema in studio può essere di tipo continuo se la risposta al fenomeno è di tipo continuo e dipende da una o più variabili indipendenti, di tipo discreto se si presenta come una sequenza di eventi rilevati in punti discreti di una variabile indipendente. |

---

(o) Si può intendere come sistema un dispositivo, una procedura od uno schema la cui funzione sia quella di operare su un'informazione, un'energia o su qualche entità fisica ed, in un intervallo di tempo opportuno, produrre una risposta sotto forma di un'informazione, un'energia o un'entità fisica.

Il primo problema che si pone in questo studio dei sistemi è quindi quello di giungere ad un modello matematico il più adeguato possibile al corrispondente fisico nei limiti del campo di funzionamento effettivo; la sua risoluzione tipica trattando di sistemi continui, è quella di uno o più sistemi differenziali che si seguono ordinatamente secondo fasi logiche. A prima vista la migliore integrazione di tali equazioni dovrebbe avvenire mediante un calcolatore analogico perchè in grado di fornire una soluzione continua, in realtà esso è capace di trattare fenomeni descrivibili con una sola variabile indipendente, mentre un sistema schematizzato da equazioni alle derivate parziali usa più variabili indipendenti ed inoltre è difficile descrivere le caratteristiche logiche del modello. Il calcolatore digitale d'altro lato discretizza il problema e rende più difficile la descrizione del sistema a causa della non aderenza dei linguaggi a sua disposizione, al problema specifico.

Da queste osservazioni sono nati due fenomeni: da un lato un nuovo tipo di calcolatore, l'ibrido che unisce le caratteristiche dell'analogico e del digitale mediante un'opportuna interfaccia, che ha così a disposizione le ampie caratteristiche logiche del digitale e la continuità e il parallelismo dell'analogico e dall'altro la costruzione sui digitali di linguaggi orientati per la simulazione di sistemi continui, capaci di simulare, entro certi limiti, il parallelismo dell'analogico, di facile apprendimento ed uso.

#### 1) Il calcolatore ibrido - Caratteristiche del hardware

Con il nome di ibrido si intende un calcolatore costituito da tre parti essenziali: un calcolatore digitale, un calcolatore analogico ed un'interfaccia, quest'ultima intesa come un dispositivo che collega fisicamente gli elementi dell'analogico con registri del digitale e consente così, attraverso opportune istruzioni del digitale stesso, la programmazione dell'analogico.

In generale il calcolatore digitale utilizzato per la costruzione di un sistema ibrido è di piccole dimensioni ossia con parola di 16 o 18 bits e ciclo di memoria oscillante attorno ad 1 usec. Le ragioni di questa scelta non sono legate all'impossibilità fisica della costruzione di calcolatori ibridi con digitali di media grandezza o grandi, ma da due fatti essenziali: la bassa precisione dello analogico, che lavora in genere su tre cifre decimali, rende inutile per molti tipi di problemi, nella parte digitale l'uso di parole di 32 bits, inoltre per il tipo di lavoro da eseguirsi sull'ibrido costituito da tentativi successivi, e prove ripetute è importante la possibilità per il programmatore di un facile accesso ad ogni parte del sistema, cosa assai più difficile su un digitale di medie o grandi proporzioni dove il software ricco e sofisticato costituisce uno sbarramento per il programmatore ed impedisce una completa maneggevolezza del sistema. E'

da notare che non occorre una configurazione particolare per costruire un ibrido, anche il sistema digitale minimo di sola unità centrale, input/output a console è sufficiente. Naturalmente una più ampia disponibilità di memorie centrali e di memoria di massa consente l'esecuzione di programmi più ampi, con possibilità di impiego del sistema molto maggiori. Meno importanti paiono le disponibilità di input/output digitali particolarmente veloci, poichè il tipo di problemi trattati non prevede la gestione di grosse quantità di dati digitali. Interessante è invece la connessione del sistema con unità di I/O di tipo analogico, che consentono, una rapida acquisizione di dati assai utili disponendo di un sistema capace di trattare l'informazione molto velocemente, ed una visualizzazione dei risultati, vedi oscilloscopi, unità video, plotters che semplifica le procedure ripetitive durante lo studio del sistema facilitando l'interazione uomo-macchina.

I calcolatori analogici impiegati nella costruzione di un ibrido sono del tipo ad amplificatori operazionali e sempre corredati della possibilità di funzionamento in ripetitivo.

Devono inoltre disporre di una espansione opportuna che permetta l'esecuzione parallela di operazioni logiche rendendo più agevole il collegamento col calcolatore digitale ovvero simulando in hardware la parte booleana del sistema in esame. In generale infatti sulla parte analogica vengano impostate le equazioni differenziali che descrivono il sistema e se il numero di elementi a disposizione come integratori, sommatore, comparatori etc. è troppo ridotto, soltanto piccoli problemi possono essere affrontati, d'altro lato se il calcolatore a disposizione è troppo grande, l'accumularsi degli errori dovuti al cattivo rendimento degli elementi è tale da falsare i risultati. Non va dimenticato inoltre che la messa a punto di un pannello analogico di dimensioni troppo vaste, seppure con l'ausilio del digitale e dei linguaggi appositamente preparati allo scopo, è molto lunga.

L'interfaccia di un sistema ibrido nella sua forma più generale può essere suddivisa in tre parti fondamentali ciascuna delle quali esegue particolari controlli e collegamenti : una interfaccia di controllo, un'interfaccia logica, ed un'interfaccia capace di trasmettere dati ad alta velocità fra l'analogico ed il digitale e viceversa.

L'interfaccia di controllo è quella parte dell'interfaccia che permette al calcolatore digitale di sostituirsi ai comandi manuali di controllo del calcolatore analogico consentendo azioni come la selezione di una ben definita console e del suo stato, il controllo dei "modi" analogici e logici del calcolatore analogico, il controllo della scala dei tempi, la selezione, la lettura di componenti analogici: amplificatori e potenziometri, l'impostazione di questi ultimi.

L'interfaccia logica è invece quella parte dell'interfaccia che trasmette informazioni logiche durante l'esecuzione del problema nei due sensi, dal digita

le all'espansione logica dell'analogico e viceversa, deve essere inoltre in grado di riconoscere ed ottemperare ad eventuali interrupts provenienti dall'analogico. La parte dell'interfacciata presiede alla trasmissione dei dati deve consentire un rapido trasferimento ed una precisa conversione di dati digitali in analogici e viceversa, sia in forma sequenziale che in parallelo. Si può osservare che per comandare l'analogico mediante il digitale attraverso l'interfacciata sono necessarie istruzioni digitali. Queste possono presentarsi sotto l'aspetto di istruzioni cablate se il digitale è stato progettato per la connessione con l'analogico o come "macro" o pseudo-istruzioni se il digitale è del tipo general-purposes e viene utilizzato anche per l'ibrido. E' chiaro che fra le due soluzioni c'è una netta differenza di velocità che può incidere nella scelta, soprattutto se l'utente desidera in qualche circostanza lavorare in tempo reale.

## 2) Il calcolatore ibrido - Caratteristiche del software

Come già accennato nel paragrafo precedente una caratteristica di un sistema ibrido è quella di essere assai scarno nel software. Questo per facilitare la costruzione di un software ad hoc per il problema che si sta risolvendo. Essenzialmente il software di un ibrido è composto da alcuni programmi di caricamento, un assembler, alcuni compilatori e traduttori. Possiede inoltre una Run-time Library<sup>(o)</sup> meno ricca ed alcune routines di I/O, qualche programma di debug (oo) più o meno sofisticato. Tutto questo bagaglio è disponibile, ma, il programmatore che lo gestisce direttamente, mancando l'azione di un supervisore in memoria. In genere è completamente affidata al programmatore la gestione degli interrupts (ooo). L'assembler si presenta pertanto come il linguaggio più indicato per sfruttare appieno le caratteristiche di un calcolatore di questo tipo. Tutte le case costruttrici forniscono però anche un Fortran capace di lavorare, attraverso l'uso di opportune subroutines, sugli elementi analogici del sistema. Si tratta in genere di un Fortran IV piuttosto modesto, ma che rispetta le convenzioni ASA. La caratteristica più tipica del software ibrido è però quella di disporre di linguaggi di tipo conversazionale tipo HYTRAN OPERATIONS INTERPRETER che consentono un lavoro a consolle sul programma sia in fase di messa a punto particolarmente del pannello analogico, sia in fase di esecuzione di simulazioni. In genere si tratta

---

(o) Run - time Library - Insieme di subroutines aritmetiche, di conversione d'ingresso/uscita che vengono caricate dal Loader insieme al programma oggetto generato dal compilatore Fortran.

(oo) Programmi di debug - Insieme di programmi che servono per la messa a punto del programma in esame, in fase di esecuzione

(ooo) Interrupt-Segnale che induce il calcolatore digitale ad interrompere il normale flusso sequenziale delle istruzioni ed a trasferire il controllo ad uno speciale programma appositamente predisposto in memoria per quella circostanza.

di linguaggi di semplice uso e di facile apprendimento che consentono rapido calcolo di espressioni aritmetiche con funzioni matematiche trascendenti, definizione e memorizzazione di variabili, memorizzazione di istruzioni e di programmi, connessioni, inserimenti e cancellatura di istruzioni, azioni di I/O, controllo logico dell'esecuzione di un programma, funzioni di salto ed iterative, controllo, indirizzamento, posizionamento degli elementi analogici e dei modi analogici. Tutte queste azioni sono richieste attraverso espressioni di forma molto-semplificata, vicina alle espressioni matematiche o al parlare comune. Si intuisce dalla struttura stessa del software ibrido, che il sistema ibrido è pensato per essere utilizzato da un programmatore esperto ossia con una buona conoscenza dell'hardware della macchina e delle tecniche di programmazione e contemporaneamente da una persona profondamente a conoscenza del problema che si sta risolvendo. Indubbiamente questo costituisce una limitazione nella sua diffusione. Difficile per il software ibrido è inoltre l'aggiornamento, perchè manca un'adeguata diffusione di questo strumento e le case costruttrici tendono a considerarlo come un'applicazione molto particolare e perciò a dedicare poche energie all'ampliamento o al miglioramento dei programmi a disposizione.

### 3) I linguaggi di simulazione dei sistemi continui

Con l'avvento dei calcolatori digitali della terza generazione è divenuta tipica una caratteristica che si era già affermata con la seconda generazione, l'hardware del sistema è del tipo general-purposes, l'orientamento del sistema per la risoluzione di problemi specifici avviene attraverso il software. Questa fondamentale premessa, unita alla velocità molto elevata, che rende molti sistemi utilizzabili per il tempo-reale fece nascere l'idea di utilizzare il digitale in sostituzione dell'analogico, simulando quest'ultimo con un software adeguato.

L'idea era in parte già sorta quando questa simulazione avveniva per la messa a punto di programmi analogici come nel caso del linguaggio APACHE, si sviluppò ulteriormente con la creazione di subroutine simulanti l'analogico ed assunse una sua veste definitiva quando nel giugno 1967 il Simulation Software Committee of Simulation Councils Inc. sancì le caratteristiche principali cui dovevano rispondere i successivi linguaggi di simulazione i quali, pensando al tipo particolare di problema cui erano rivolti, vennero definiti linguaggi di simulazione di sistemi continui Continuous System Simulation Languages (CSSL). Questi linguaggi offrono alcuni indiscutibili vantaggi. Per la loro semplicità di programmazione sono facilmente accessibili senza necessità di utilizzare programmatori particolarmente esperti, permettono di utilizzare nella simulazione configurazioni analogiche anche molto grandi, possono interagire con il linguaggio procedurale tipico del calcolatore in uso o il Fortran, o l'Algol o il PL/I.

Complemento essenziale ai fini dell'utilizzazione dei linguaggi di simulazione con calcolatori digitali in sostituzione dei sistemi ibridi è la disponibili-

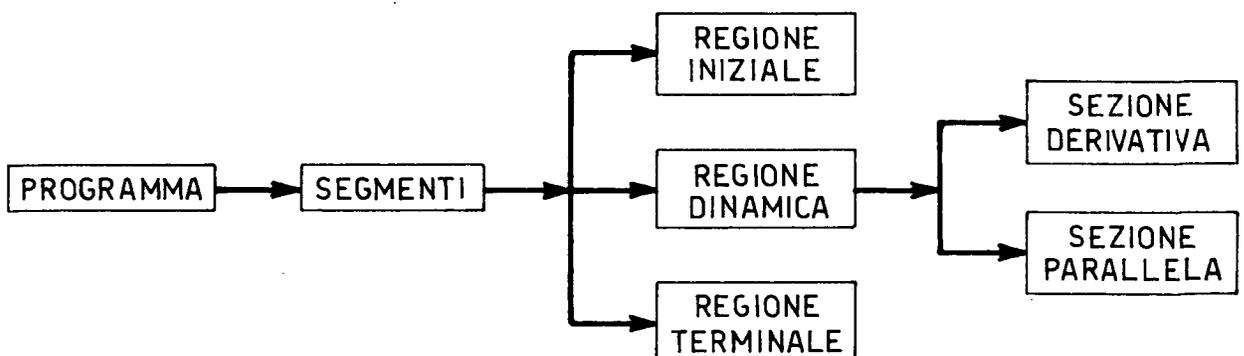
lità dei periferici di visualizzazione che consentano l'interazione operatore/dinamica del sistema, oltre all'attitudine converzionale del sistema operativo in funzione sul calcolatore.

#### 4) Caratteristiche essenziali del CSSL

Il CSSL è stato sviluppato da un comitato composto dagli autori dei linguaggi precedentemente a disposizione per la simulazione di sistemi continui e cioè principalmente il DSL/90, MIMIC, MIDAS, PACTOLUS ed altri. Essenzialmente gli obiettivi di questo linguaggio sono: una facile forma per la descrizione del modello, la possibilità con pochi operatori di operare su un gran numero di problemi che coinvolgono equazioni differenziali, buone subroutines d'integrazione incorporate, possibilità di agire sui parametri durante l'esecuzione della simulazione, un buon assieme di routines diagnostiche sia per la fase di compilazione che per quella di esecuzione.

In genere le istruzioni di tale linguaggio sono costruite facendo riferimento ai precedenti linguaggi di simulazione MIMIC e DSL/90 e ai linguaggi procedurali tipo Fortran in particolare, auspicandone la sostituzione con il PL/I. Si presentano come tipiche altre due caratteristiche di un linguaggio rispondente alle specifiche del CSSL: la flessibilità ossia il programmatore può programmare la simulazione del suo sistema fin nei minimi dettagli ad esempio mediante la struttura esplicita dell'SL/1, la possibilità di introdurre nuovi operatori attraverso la costruzione di "macro" ossia di un insieme di istruzioni che sono inserite nel programma ogni volta che la "macro" viene richiamata per nome. E' da osservare che tuttavia le specifiche prevedono di conservare come alternativa una struttura vicina a quella a blocchi dei linguaggi tipo MIDAS, SCADS, CSMP/1130 di facile programmazione per un utente con preparazione di tipo analogico. In tale tipo di linguaggio la programmazione avviene infatti attraverso un diagramma a blocchi eguale a quello analogico, in cui gli elementi fisici dell'analogico, sono sostituiti da subroutines che simulano il loro funzionamento sul digitale.

Il CSSL prevede nella sua forma più ampia che il programma sia strutturato secondo il seguente schema :



Il "programma" costituisce l'analogo del programma principale di un linguaggio procedurale ed i segmenti che possono essere numerosi quanto si vuole, rappresentano ciascuno una simulazione completa e si presentano come l'equivalente della subroutine di un linguaggio procedurale. La logica che ordina l'esecuzione di tali segmenti è programmata in un linguaggio procedurale. Ciascun segmento è costituito da tre sezioni. La regione iniziale che comprende tutte le operazioni di inizializzazione e di I/O che debbono essere predisposte prima di una simulazione. In questa sezione è compresa una routine interpretativa che permette, in fase di esecuzione, le interazioni fra il programmatore e il programma, consentendo ad esempio la variazione dei parametri, la lettura del valore di una variabile durante la simulazione, il controllo e la modifica dei parametri di integrazione. La regione dinamica è quella parte della simulazione che interagisce con l'esterno. Essa rappresenta ad ogni passo tutti i calcoli e le operazioni di I/O ad ogni valore discreto della variabile indipendente. Essa è suddivisa in due parti una sezione derivativa, che descrive il sistema dinamico che deve essere simulato, ossia calcola la derivata della variabile di stato per un ben determinato valore della variabile indipendente ed una sezione parallela che ha le stesse funzioni descrittive della precedente, ma in cui non sono mai eseguite integrazioni. Infine la sezione terminale che contiene tutte le istruzioni relative alla conclusione di una simulazione ed al ritorno del controllo all'inizio della simulazione stessa. Si pensa che un linguaggio rispondente alle specifiche CSSL debba disporre di istruzioni che definiscano le caratteristiche dei blocchi, ossia delle strutture tipo segmenti, regioni, sezioni, tali istruzioni vengono definite di struttura. Inoltre deve poter usufruire di istruzioni descrittive del sistema da simulare che possono essere o di tipo particolare o utilizzare le istruzioni del linguaggio procedurale connesso; ed infine istruzioni di controllo sia della fase di compilazione che di esecuzione. Naturalmente gran parte delle istruzioni del linguaggio procedurale di riferimento sono riconoscibili ed eseguibili da parte del linguaggio di simulazione. Nucleo fondamentale di un linguaggio di simulazione di sistemi continui è il sistema d'integrazione. Le specifiche CSSL prevedono che tale sistema disponga di due ingressi principali uno per la inizializzazione ed uno per l'integrazione vera e propria. Nella fase di inizializzazione il sistema di integrazione pone le condizioni iniziali, richiama l'appropriato e prescelto algoritmo d'integrazione e dispone in memoria un'opportuna tabella sulla storia passata del sistema, se la procedura d'integrazione lo richiede.

Nella fase d'integrazione si comunica alla routine di integrazione le specifiche della sezione derivativa interessata e l'intervallo di comunicazione dell'integrazione, ossia l'intervallo in cui i risultati dell'integrazione, possono essere conosciuti all'esterno, prescindendo dal passo d'integrazione usato.

Gli algoritmi a disposizione del programmatore debbono essere parecchi,

particolarmente raccomandati sono il metodo di Eulero e il metodo di Runge-Kutta fra le tecniche ad un solo passo, ed i metodi di Adams per le tecniche a più passi del tipo predictor-conector.

Esempi di linguaggi a disposizione che rispondono a queste specifiche sono l'SL-1, disponibile sui calcolatori SIGMA5 e SIGMA7 della XDS, ed il CSMP/360 disponibile sulla serie 360 della IBM.

#### 5) Confronto fra la simulazione ibrida e quella digitale

Un confronto fra la simulazione ibrida e quella digitale alla luce di quanto prima detto sulle rispettive caratteristiche deve essere fatto distinguendo due aspetti: uno di tipo tecnico scientifico ed uno di natura economica.

Il grande vantaggio che si può ottenere dall'uso di un calcolatore analogico è dato dalla possibilità di trattare più variabili con continuità ed in parallelo con tempi di risposta rapidissimi, seppure con una precisione abbastanza bassa. Questo vantaggio viene sfruttato nella simulazione ibrida per rappresentare quelle parti del modello che sono esprimibili analiticamente. Per descrivere caratteristiche qualitative, per esprimere decisioni, ci avvale invece del calcolatore digitale. Si vede così che il calcolatore ibrido consente di utilizzare il trattamento continuo dell'informazione proprio dell'analogico, di raffinare attraverso la maggior precisione, tipica del digitale, i risultati ottenuti e di esprimere con completezza tutte quelle caratteristiche del sistema che non sono di natura strettamente analitica. Questo è un aspetto particolarmente importante perchè l'utilità di una simulazione dipende, in forma essenziale, dalla bontà del modello costruito e dalla sua coincidenza con il sistema reale in esame. D'altro canto il calcolatore ibrido, a causa delle limitazioni costruttive dell'analogico può perdere validità per la simulazione di sistemi di grandi dimensioni. Due casi possono essere citati come casi in cui l'utilizzazione di tecniche ibride sia particolarmente felice:

- 1) Simulazione di sistemi da ripetersi un gran numero di volte. La soluzione di problemi di questo tipo richiede sul digitale tempi molto più lunghi.
- 2) Simulazione di sistemi con un'ampia banda di frequenza. Ad esempio lo studio di una trasformazione che duri 10 sec. in un sistema avente frequenze principali 1 e per esempio 1 kHz, comporta una quantizzazione sul tempo almeno pari a  $10^5$
- 3) Qualunque tipo di sistema in cui la rapidità del trattamento dell'informazione sia essenziale. L'analogico può infatti soddisfare al concetto di tempo reale in forma più vicina all'evento reale che non il digitale.

I vantaggi di natura tecnica si presentano numerosi, non così forse quelli di natura economica. L'uso del calcolatore ibrido è infatti assai laborioso, richiede una buona conoscenza del calcolatore e della programmazione sia analogica che digitale, oltre una profonda conoscenza del problema da risolvere. Esclude perciò dai possibili utilizzatori i programmatori professionisti, ma richiede una

grossa preparazione di programmazione all'utilizzatore-progettista. La procedura di lavoro sul calcolatore ibrido è di tipo tipicamente conversazionale, con lunghissime fasi di messa a punto e prova è quindi una tecnica molto dispendiosa perchè il calcolatore viene bloccato per tempi lunghi su un unico problema. Va infine notato che un calcolatore ibrido può essere utilizzato soltanto per problemi di simulazione.

Differente si presenta la posizione dei linguaggi di simulazione. Da un punto di vista tecnico, essi non hanno alcuni vantaggi tipici dell'analogico ossia la continuità ed, in gran parte, il parallelismo, benchè quest'ultimo venga in parte simulato. Si presentano quindi essenzialmente come strumenti atti a semplificare la stesura di un programma che identifichi un modello. E' chiaro che conservano tutti i vantaggi tipici della struttura digitale. Sono particolarmente indicati per due tipi di problemi:

1) Simulazione di sistemi di grandi dimensioni : basta infatti disporre di un calcolatore sufficientemente grande, mentre formalmente non esiste nessuna complessità maggiore nella programmazione e messa a punto del modello;

2) Simulazione di sistemi di dimensioni piccole o medie con un gran numero di non-linearità, o con non-linearità non usuali, o con ritardi di trasporto.

Da un punto di vista economico si presentano in una posizione di netto vantaggio. I linguaggi infatti tendono ad una forma molto semplice ed il più vicino possibile, compatibilmente con la generalità del linguaggio stesso, al costruttore del modello, richiedono così all'utente un minimo di capacità di programmazione. La simulazione inoltre viene eseguita su un calcolatore digitale "general-purposes" di grandi dimensioni, talvolta in "time-sharing" esso pertanto non viene sotto nessun aspetto monopolizzato dal problema in esame.

A conclusione occorre osservare che tutte queste osservazioni sono suscettibili di ampie e rapide variazioni, perchè entrambi i problemi sia quello della costruzione di calcolatori ibridi, sia quello dell'uso di linguaggi di simulazione sono in sviluppo, necessitando il primo della costruzione di un software adeguato ed i secondi del raggiungimento di una struttura sempre più generale e flessibile.

## Bibliografia

1. Strauss, J. et al "The SCI Continuous System Simulation Language (CSSL)" Simulation, Vol. 9 No. 6, (1967)
2. XDS Sigma 5/7 SL-1 Reference Manual # 901676
3. System/360 Continuous System Modeling Program (360-CX-16X) System Manual; IBM Y20-0111-0
4. Schram S. and Chai S. "Simulation with SL-1" Proceedings of the Summer Computer Simulation Conference, 1970 - Denver - Colorado
5. T. Utsumi et al. "SWAPSO Development : Optimization of CSMP/360 or Hybrid Mathematical Models" Proceedings of the Summer Computer Simulation Conference, 1970 - Denver - Colorado
6. Bekey G.A. and Karplus W.J. "Hybrid Computation" John Wiley (1968)
7. Rullgard, A. "Some aspects of hybrid Computers" Simulation, Vol. 11, No 4, (1968)
8. Korn A.G. "Digital-computer interface systems" Simulation, Vol. 11, No 6, (1968)
9. Tiechroew, D. et al. "Discussion of computer simulation techniques and comparison of languages" Simulation Vol. 9 No. 4 (1967)
10. The E.A.I. 590/690 Hybrid Computing System - Reference Manuals
11. The PDP-15/EAI 680 Hybrid Computing System - Reference Manuals

NOTA

Il presente articolo è stato redatto nell'ambito di attività svolte presso la Stanford University, California, ove gli autori usufruivano di borse conferite rispettivamente dal Comitato Nazionale per le Scienze Matematiche del C.N.R. e dall' E.S.R.O. - N.A.S.A. International Fellowship Programme.